

[首页](#)[官方教程](#)[视频课程](#)[社区开源](#)[正在开发中.....](#)[登录](#)

Spring Data JPA系列：使用Sort进行排序（Using Sort）

- 通过上一节的学习，我们知道了如何用@Query注解来实现灵活的查询。在上一节的示例中，我也尝试给出简单的排序，通过JPQL语句以及原生SQL来实现的。这样的实现，虽然在一定程度上可以应用，但是灵活度不够，因此结合@Query注解，我们可以使用Sort来对结果进行排序。**

1、在CustomerRepository内添加方法

```
/**
 * 一个参数，匹配两个字
 * @param name2
 * @param sort 指定排序的参数，可以根据需要调整
 * @return
 * 这里Param的值和=:后面的参数匹配，但不需要和方法名对应的参数值对应
 */
@Query("select c from Customer c where c.firstName=:name or c
List<Customer> findByName4(@Param("name") String name2,
```

方法一如既往，是声明式的，只是在原有方法的基础上，加上Sort（`org.springframework

2、在CustomerController中测试

```
/**
 * @Query注解方式查询，
 * 用@Param指定参数，匹配firstName和lastName
 */
@RequestMapping("/findByName")
public void findByName4(){
    //按照ID倒序排列
    System.out.println("直接创建sort对象，通过排序方法和属性名");
    Sort sort = new Sort(Sort.Direction.DESC,"id");
    List<Customer> result = repository.findByName4("Bau
    for (Customer customer:result){
        System.out.println(customer.toString());
    }
    System.out.println("-----");
    //按照ID倒序排列
    System.out.println("通过Sort.Order对象创建sort对象");
    Sort sortx = new Sort(new Sort.Order(Sort.Direction.DESC,
    List<Customer> resultx = repository.findByName4("Ba
    for (Customer customer:result){
        System.out.println(customer.toString());
    }
    System.out.println("-----");

    System.out.println("通过排序方法和属性List创建sort对象");
    List<String> sortProperties = new ArrayList<>();
    sortProperties.add("id");
    sortProperties.add("firstName");
    Sort sort2 = new Sort(Sort.Direction.DESC,sortProperties)
    List<Customer> result2 = repository.findByName4("Ba
    for (Customer customer:result2){
        System.out.println(customer.toString());
    }
}
```

[Spring Data](#)

阅读	评论	喜欢
240	0	0

[喜欢](#)[SpringForAll](#)

文章	问答	粉丝
79	6	236

[关注](#)

```
System.out.println("-----");

System.out.println("通过创建Sort.Order对象的集合创建sort对象");
List<Sort.Order> orders = new ArrayList<>();
orders.add(new Sort.Order(Sort.Direction.DESC, "id"));
orders.add(new Sort.Order(Sort.Direction.ASC, "firstName"));
List<Customer> result3 = repository.findByName4("Ba
for (Customer customer:result3){
    System.out.println(customer.toString());
}
System.out.println("-----");
}
```

这里总共列举了四种排序方式：

- 1) 直接创建Sort对象，适合对单一属性做排序
- 2) 通过Sort.Order对象创建Sort对象，适合对单一属性做排序
- 3) 通过属性的List集合创建Sort对象，适合对多个属性，采取同一种排序方式的排序
- 4) 通过Sort.Order对象的List集合创建Sort对象，适合所有情况，比较容易设置排序方式

对应着我们的使用场景来进行选择创建Sort对象的方式。

注意，这里并没有列举所有的Sort使用方式，还有忽略大小写，使用JpaSort.unsafe、聚合函数等进行排序，查询的属性值是Entity的属性名，不是数据库的字段，要注意到!!

参考：

官方文档：<https://docs.spring.io/spring-data/jpa/docs/current/reference/html>

API官方文档：<http://docs.spring.io/spring-data/data-jpa/docs/current/api/>

JPQL文档：<http://www.blogjava.net/calmJava/archive/2011/04/01/347450.html>

DEMO示例：<https://github.com/icnws/spring-data-jpa-demo>

标签：

Spring Data

▼ 评论

快来抢沙发吧！



提交

JET
BRAINS

Level up your code
with a
Pro Java IDE

IntelliJ IDEA

Get it now